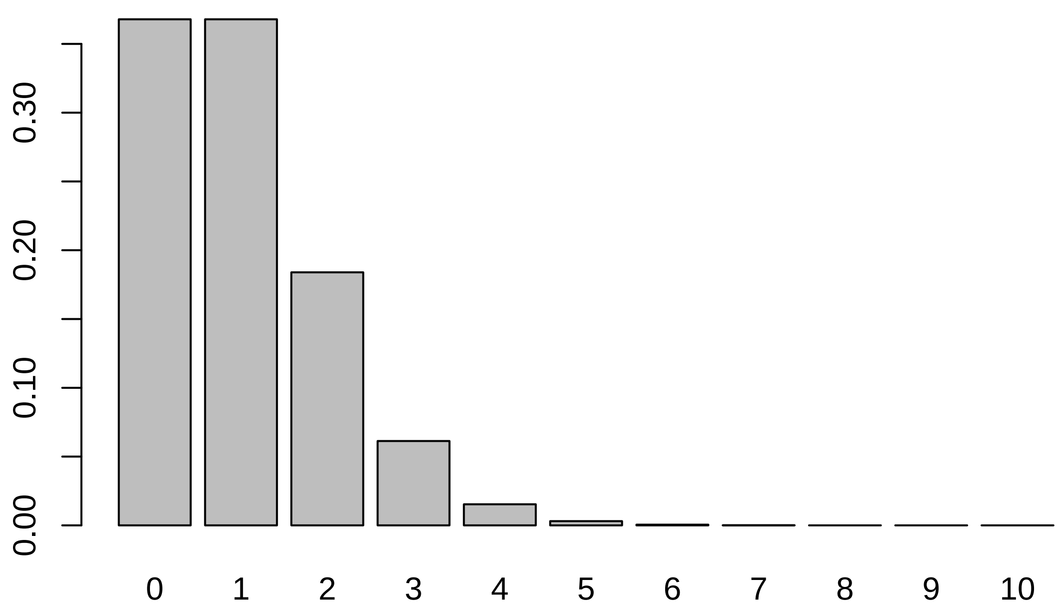
DiscreteProbabilityDistributions.R

Administrator

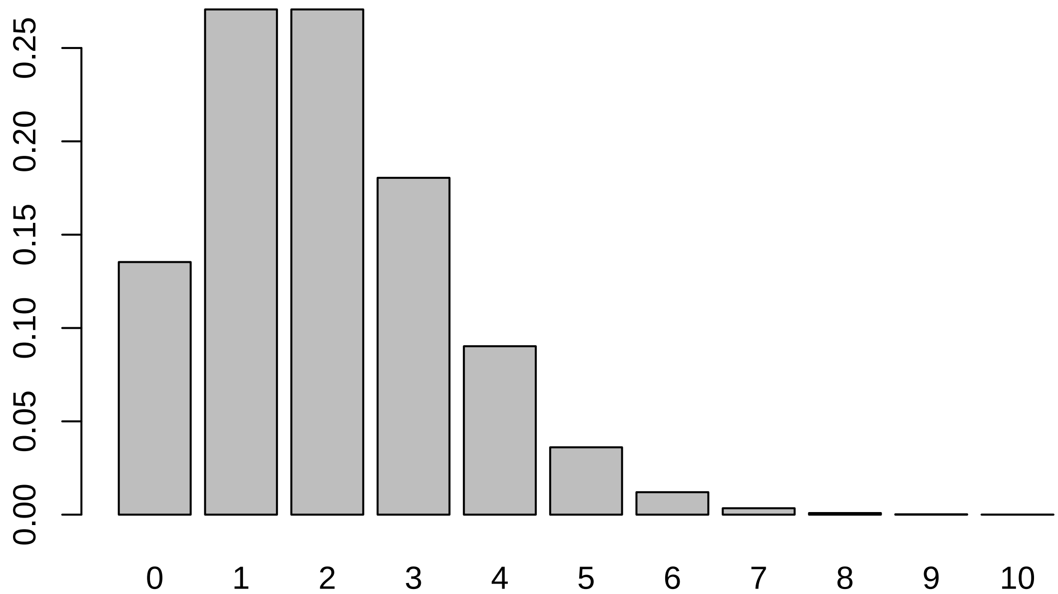
Tue Mar 22 14:51:05 2016

```r
# Discrete Probability distributions and generating functions
# 18 March 2016
# NJG
```

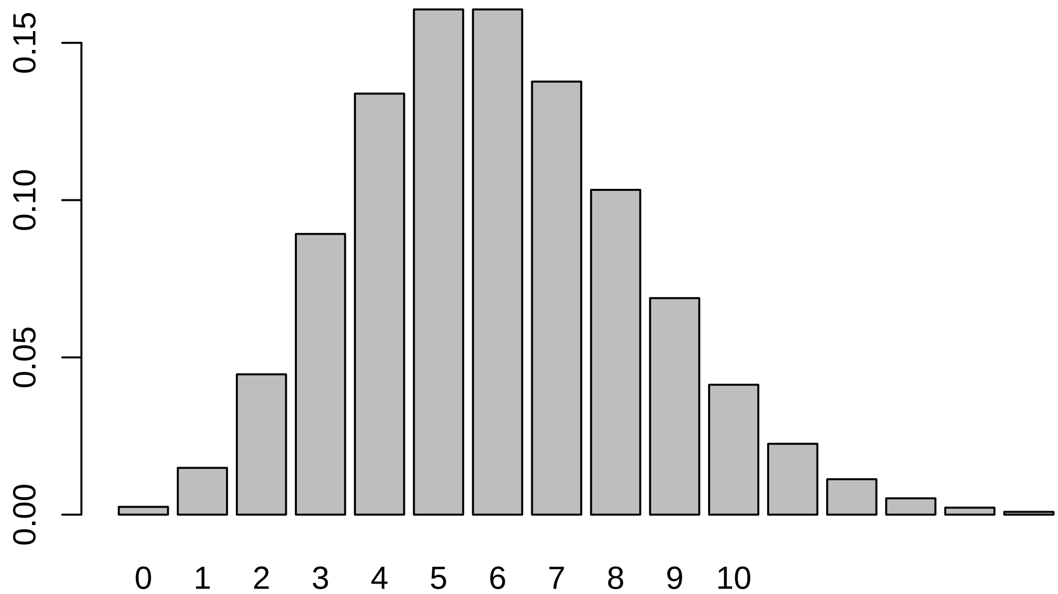
```r
# Poisson distribution
# Discrete X >= 0
# Random events with a constant rate lambda
# (observations per time or per unit area)
# Parameter lambda > 0

# "d" function generates probability density

MyVec <- dpois(x=seq(0,10),lambda=1)
names(MyVec) <- seq(0,10)
barplot(height=MyVec)
```



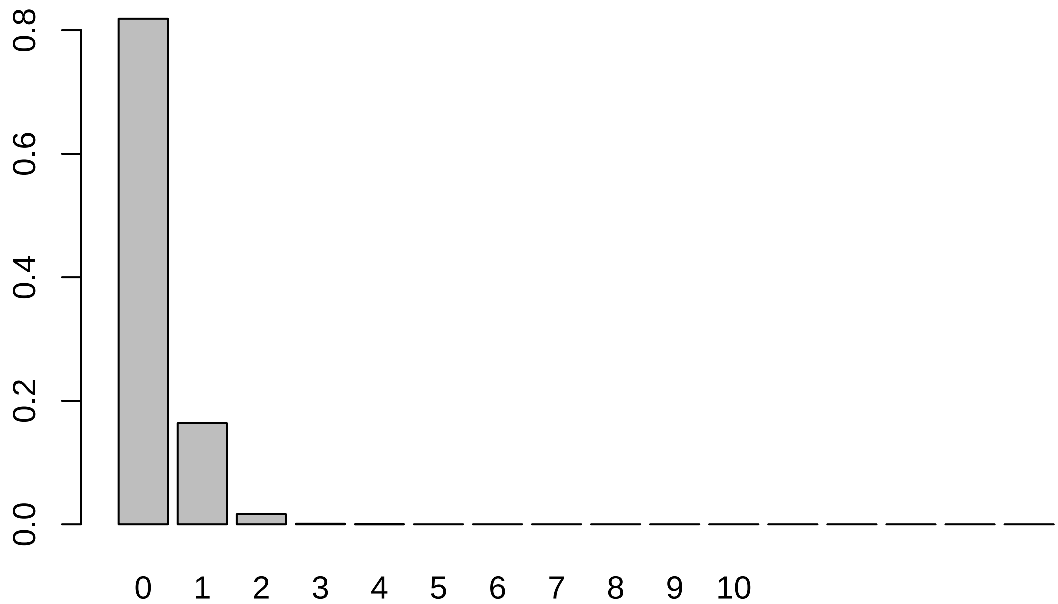

```r
MyVec <- dpois(x=seq(0,10),lambda=2)
names(MyVec) <- seq(0,10)
barplot(height=MyVec)
```




```
MyVec <- dpois(x=seq(0,15),lambda=6)
names(MyVec) <- seq(0,10)
barplot(height=MyVec)
```




```
MyVec <- dpois(x=seq(0,15),lambda=0.2)
names(MyVec) <- seq(0,10)
barplot(height=MyVec)
```



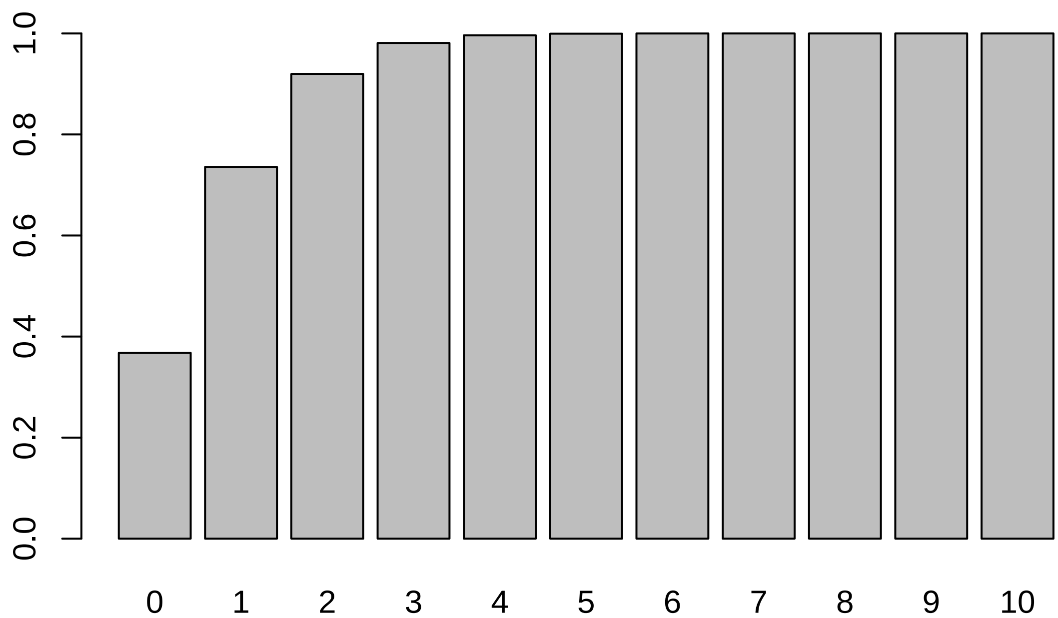

```r
sum(MyVec)  # sum of density function = 1.0 (total area under curve)
```

```
## [1] 1
```

```r
# for a Poisson distribution with lambda=2,
# what is the probability that a single draw will yield X=0?

dpois(x=0,lambda=2)
```

```
## [1] 0.1353353
```

```r
# "p" function generates cumulative probability density; gives the
# "lower tail" cumulative area of the distribution

MyVec <- ppois(q=seq(0,10),lambda=1)
names(MyVec) <- seq(0,10)
barplot(height=MyVec)
```



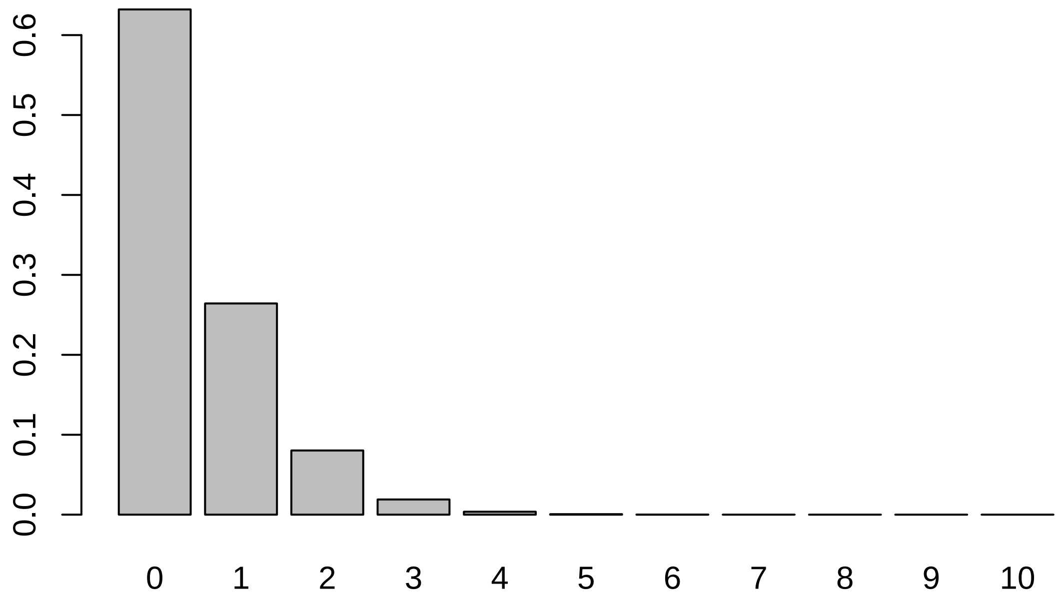

```r
# for a Poisson distribution with lambda=2,
# what is the probability that a single random draw will yield X <= 1?

ppois(q=1, lambda=2)
```

```
## [1] 0.4060058
```

```r
# to generate the "upper tail", use lower.tail=FALSE
# important: lower tail P(X <= z) but upper tail P(X > z)

# Therefore for any z, P(lower.tail z) + P(upper.tail z) = 1.0
z <- 3
ppois(q=3,lambda=1) + ppois(q=3,lambda=1,lower.tail=FALSE)
```

```
## [1] 1
```

```r
MyVec <- ppois(q=seq(0,10),lambda=1,lower.tail=FALSE)
names(MyVec) <- seq(0,10)
barplot(height=MyVec)
```




```r
# for a Poisson distribution with lambda=2,
# what is the probability that a single random draw will yield X >= 4?

ppois(q=3,lambda=2,lower.tail=FALSE)
```

```
## [1] 0.1428765
```

```r
# Because the total area under the curve is 1.0,
# we can subtract the two tail probabilities from 1.0
# to obtain the probability of a value being in a certain interval.

# for a Poisson distribution with lambda = 4,
# what is the probability that a single random draw
# will yield a value X that is between 1 and 3?

lowerTail <- ppois(q=0,lambda=4)
upperTail <- ppois(q=3,lambda=4, lower.tail=FALSE)

IntervalP <- 1.0 - lowerTail - upperTail
print(IntervalP)
```

```
## [1] 0.4151545
```

```r
# Lets check this by calculating and summing
# the probabilities for the individual integer values

x1 <- dpois(x=1,lambda=4)
x2 <- dpois(x=2,lambda=4)
x3 <- dpois(x=3,lambda=4)

sum(x1,x2,x3)
```

```
## [1] 0.4151545
```

```r
# the "q" quantile function is the "inverse" of the "p" function.
# We give a tail probability and it gives us a
# value X that would correspond to that.

# the quantile is right continuous
# qpois(p, lambda) is the smallest integer such that P(X <= z) >= p.

qpois(p=0.75,lambda=3)
```

```
## [1] 4
```

```r
# The q function can be used to generate a 95% interval on the distribution:

qpois(p=c(0.025,0.975),lambda=10)
```

```
## [1]  4 17
```

```r
qpois(p=0.975,lambda=10,lower.tail=TRUE)
```

```
## [1] 17
```

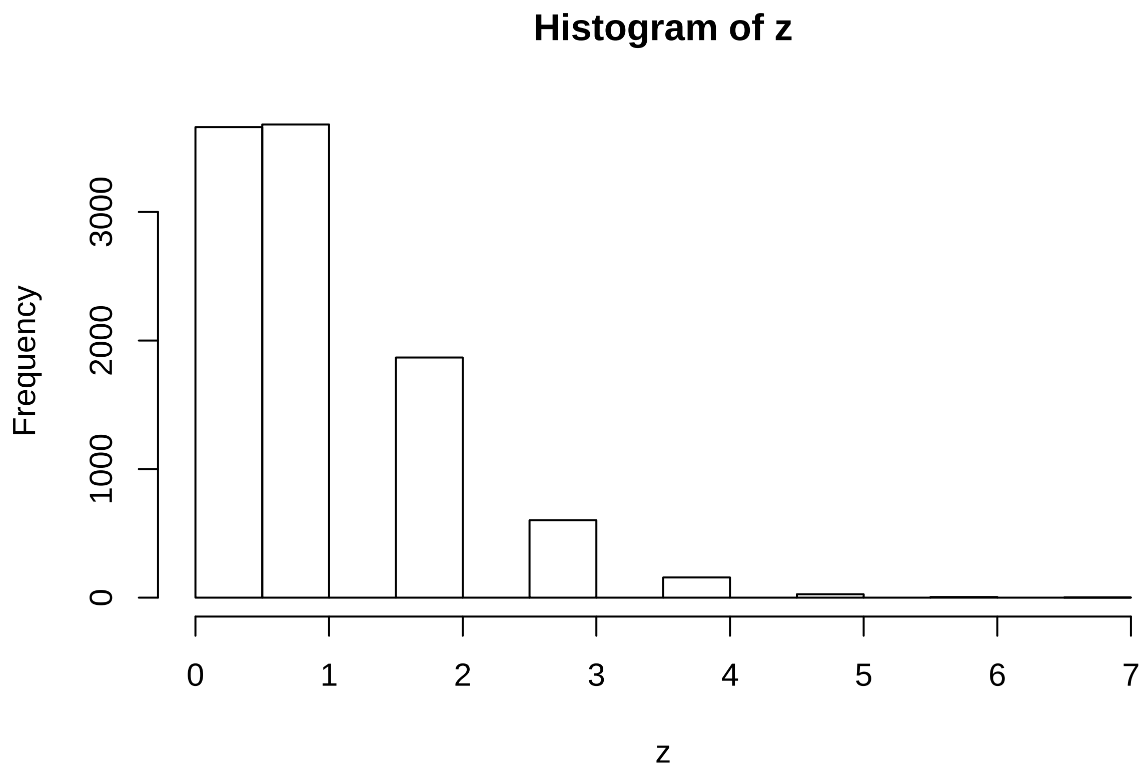
```r
# We can also find the midpoint of probability in the distribution:

qpois(p=0.5,lambda=3)
```

```
## [1] 3
```

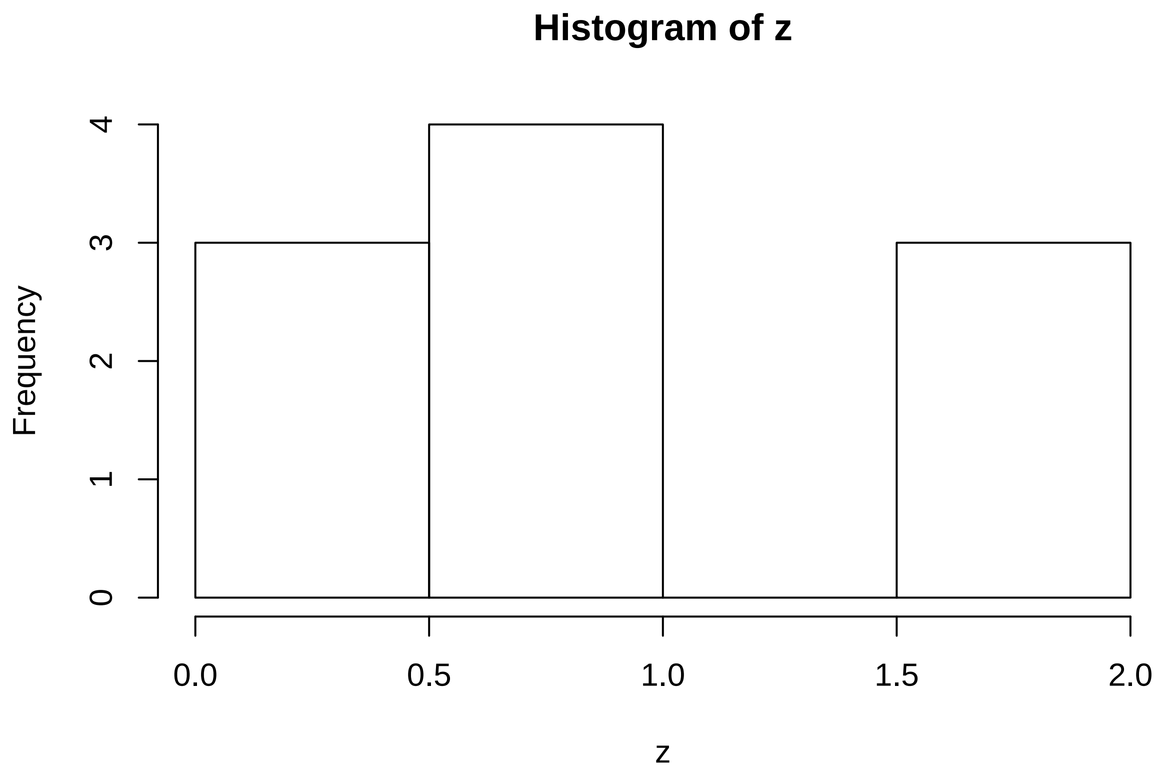
```r
qpois(p=0.5,lambda=0.1)
```

```
## [1] 0
```

```r
# Use rpois to generate a random set of values from a distribution:

z <- rpois(n=10000,lambda=1)
hist(z,breaks=10)
```




```r
z <- rpois(n=10,lambda=1)
hist(z)
```




```
# Make comparisons of sample values with predicted values
z <- rpois(n=10000,lambda=1) # very large sample
table(z)                     # get frequencies
```

```
## z
##    0    1    2    3    4    5    6    7
## 3758 3593 1867  601  136   36    8    1
```

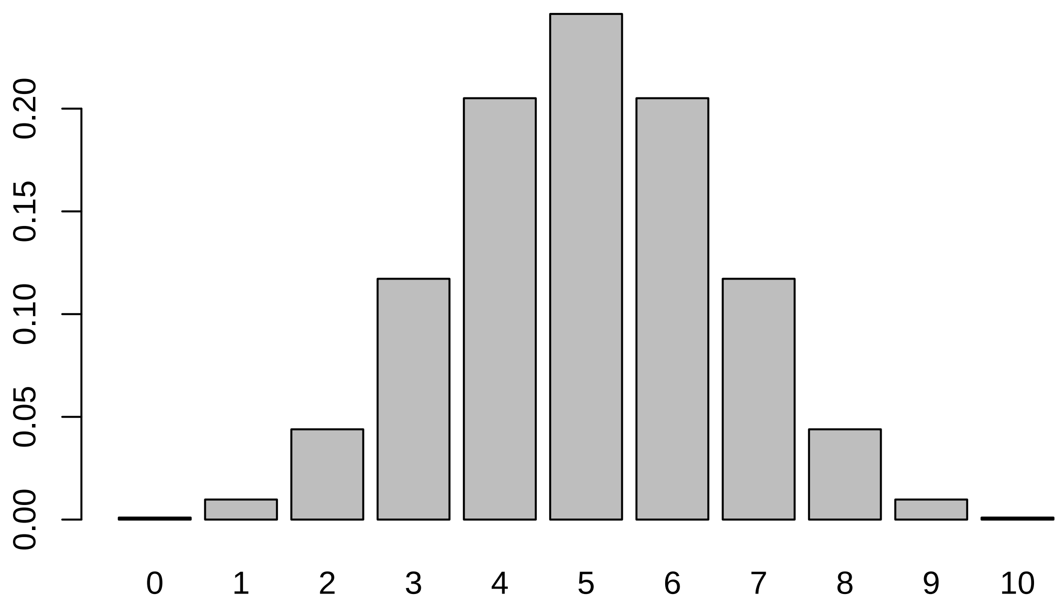
```
mean(z==5)                   # get proportion of values = 5
```

```
## [1] 0.0036
```

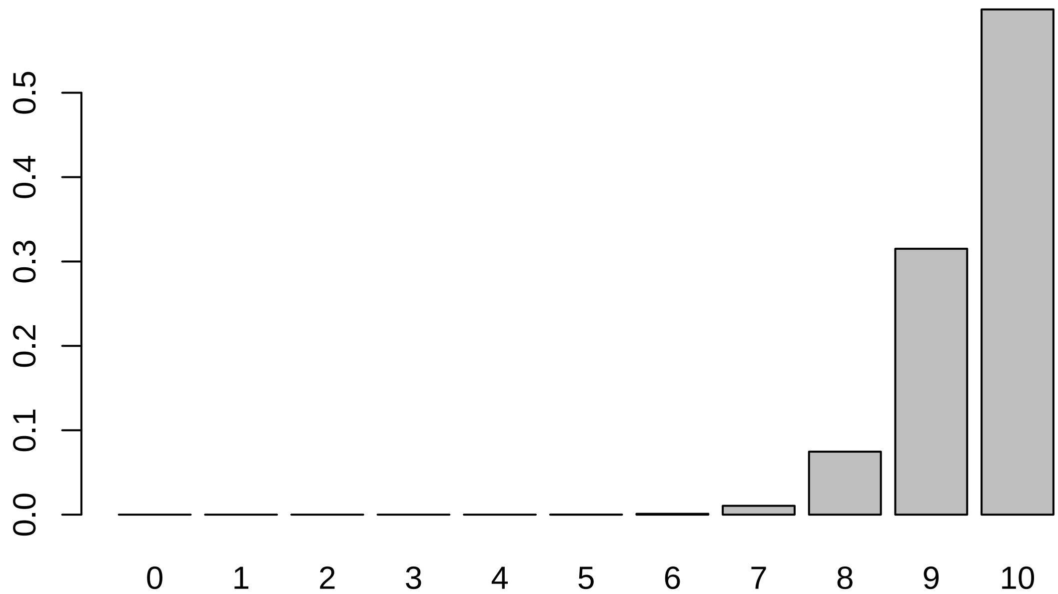
```
dpois(x=5,lambda=1)          # get theoretical proportion
```

```
## [1] 0.003065662
```

```
# Binomial distribution
# p = probability of a dichotomous outcome
# size = number of trials
# x = possible outcomes

# use "d" binom for density function
MyVec <- dbinom(x=seq(0,10),size=10,prob=0.5)
names(MyVec) <- seq(0,10)
barplot(height=MyVec)
```




```r
MyVec <- dbinom(x=seq(0,10),size=10,prob=0.95)
names(MyVec) <- seq(0,10)
barplot(height=MyVec)
```




```r
# use "p" binom for cumulative distribution

# what is probability of getting 5 heads out of 10 tosses?
dbinom(x=5,size=10,prob=0.5)
```

```
## [1] 0.2460938
```

```r
# what is the probability of getting 5
# or fewer heads out of 10 tosses?
pbinom(q=5,size=10,prob=0.5)
```

```
## [1] 0.6230469
```

```r
pbinom(q=4,size=9,prob=0.5)
```

```
## [1] 0.5
```

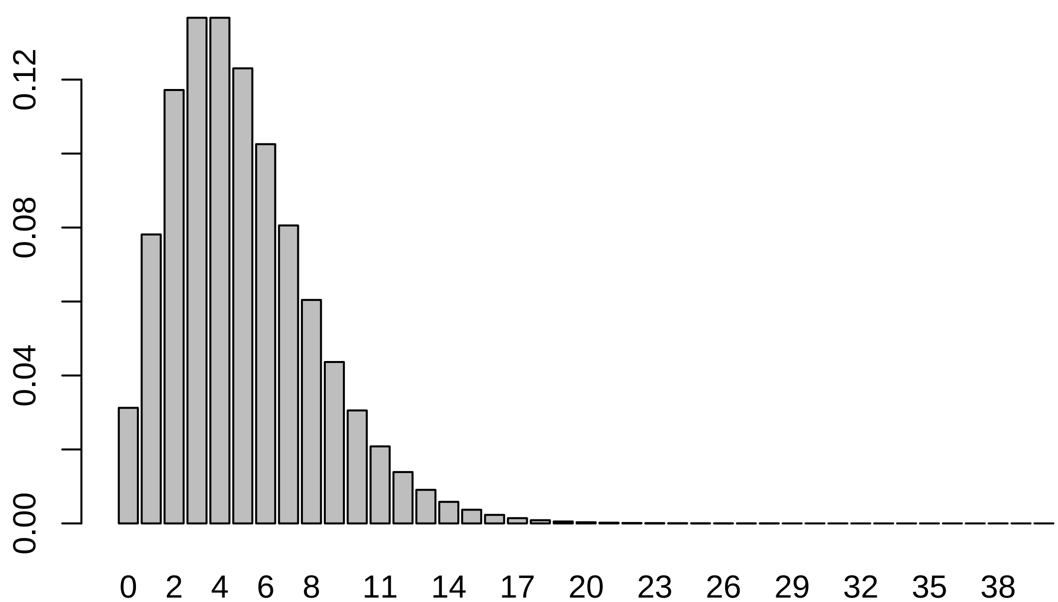
```r
# use "q" binom for quantiles

# what minimum number of heads will be found
# for 40% of 50 trials with p = 0.5?

qbinom(p=0.4,size=50,prob=0.5)
```

```
## [1] 24
```

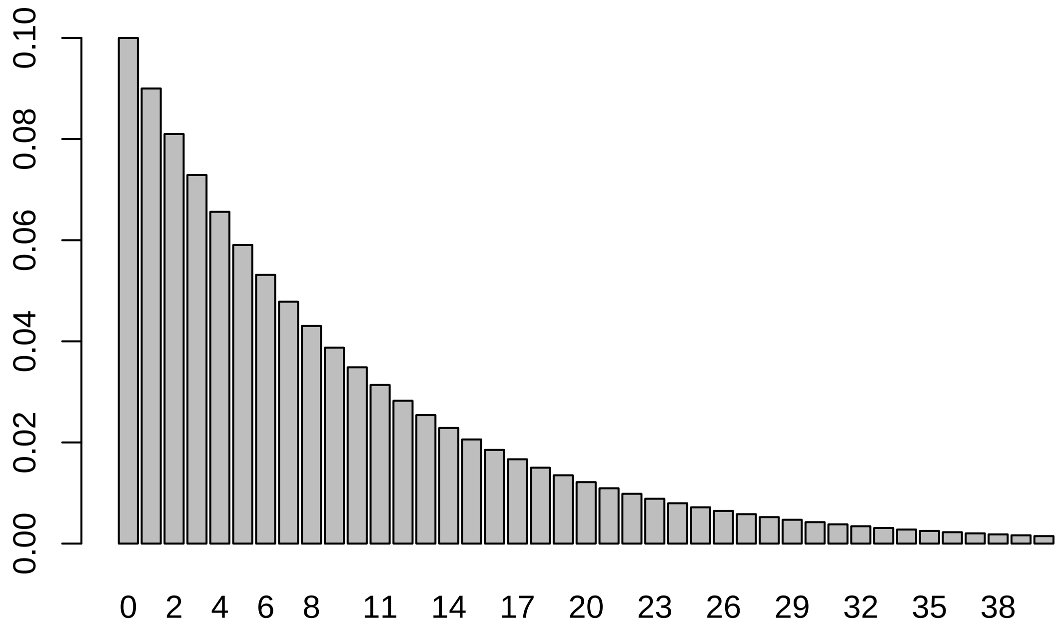
```r
# what is a 95% confidence interval for 100 trials
# of a coin with p = 0.7 for heads?
qbinom(p=c(0.025,0.975),size=100,prob=0.7)
```

```
## [1] 61 79
```

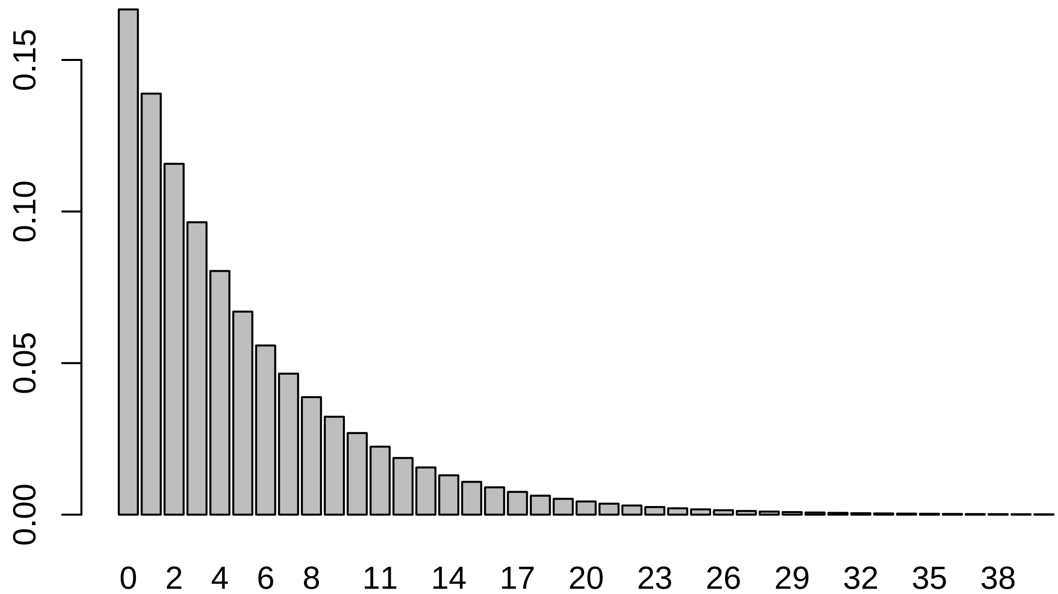
```r
# negative binomial: number of failures (values of MyVec)
# in a series of (Bernouli) with p=probability of success
# before a target number of successes (= size)
# generates a discrete distribution that is more
# heterogeneous ("overdispersed") than Poisson
MyVec <- dnbinom(x=seq(0,40), size=5, prob=0.5)
names(MyVec) <- seq(0,40)
barplot(height=MyVec)
```




```r
# geometric series is a special case where N= 1 success
# each bar is a constant fraction 1 - "prob" of the bar before it
MyVec <- dnbinom(x=seq(0,40), size=1, prob=0.1)
names(MyVec) <- seq(0,40)
barplot(height=MyVec)
```




```r
# alternatively specify mean = mu of distribution and size,
# the dispersion parameter (small is more dispersed)

MyVec <- dnbinom(x=seq(0,40),size=1,mu=5)
names(MyVec) <- seq(0,40)
barplot(height=MyVec)
```




```
# also have the "pnbinom", "qnbinom" and "rnbinom" functions
# Probability of drawing a 3 or smaller from a negative binomial:
pnbinom(q=3,size=1,mu=5)
```

```
## [1] 0.5177469
```

```
# 5 percent lower value for a negative binomial
qnbinom(p=0.05,size=10,mu=5)
```

```
## [1] 1
```

```
# 95% confidence interval for a geometric series
qnbinom(p=c(0.025,0.975),prob=0.5,size=10)
```

```
## [1]  3 20
```

```
# random sample from a negative binomial

MyVec <- rnbinom(n=1000,size=1,mu=20)
quantile(MyVec,prob=c(0.025,0.975))
```

```
##  2.5% 97.5%
##  0.00 88.05
```

```r
# compare to exact calculation
qnbinom(p=c(0.025,0.975),size=1,mu=20)
```

```
## [1]  0 75
```

```r
# Sampling from a vector of elements (not necessarily numbers)

Species <- LETTERS
print(Species)
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q"
## [18] "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
```

```r
# Sampling with or without replacement
# Sampling equiprobably or with specified (unequal) probabilities)

# Case 1: Sampling equiprobably with replacement

sample(Species,size=10,replace=TRUE)
```

```
##  [1] "M" "V" "R" "H" "R" "K" "Y" "Y" "B" "I"
```

```r
# Case 2: Sampling equiprobably without replacement

sample(Species,size=10,replace=FALSE)
```

```
##  [1] "D" "O" "R" "V" "C" "Z" "G" "E" "P" "M"
```

```r
# Case 3: Sampling with specified probabilities with replacement

# any set of non-negative numbers
# that is the same length as the vector)
MyProbs <- seq(1,26)

sample(Species, size=10,replace=TRUE,prob=MyProbs)
```

```
##  [1] "N" "V" "M" "R" "S" "N" "T" "W" "Y" "D"
```

```r
# Case 4: Sampling with specified probabilities, without replacement

sample(Species, size=10,replace=FALSE,prob=MyProbs)
```

```
##  [1] "D" "S" "Y" "P" "R" "O" "Q" "E" "X" "Z"
```